QSM368ZP-WF&SG368Z 系列
Linux&Ubuntu Secure Boot 应用指导

智能产品

版本：1.1

日期：2026-01-09

状态：受控文件

上海移远通信技术股份有限公司（以下简称“移远通信”）始终以为客户提供最及时、最全面的服务为宗旨。如需任何帮助，请随时联系我司上海总部，联系方式如下：

上海移远通信技术股份有限公司
上海市松江区泗泾镇外婆泾路 8 号　邮编：201601
电话：+86 21 5108 6236　　邮箱：info@quectel.com

或联系我司当地办事处，详情请登录：https://www.quectel.com.cn/contact。

如需技术支持或反馈我司技术文档中的问题，请随时登录网址：
https://www.quectel.com.cn/contact?tab=t 或发送邮件至：support@quectel.com。

前言

移远通信提供该文档内容以支持您的产品设计。您须按照文档中提供的规范、参数来设计产品。同时，您理解并同意，移远通信提供的参考设计仅作为示例。您同意在设计您目标产品时使用您独立的分析、评估和判断。在使用本文档所指导的任何硬软件或服务之前，请仔细阅读本声明。您在此承认并同意，尽管移远通信采取了商业范围内的合理努力来提供尽可能好的体验，但本文档和其所涉及服务是在“可用”基础上提供给您的。您知悉并同意，移远通信可在未事先通知的情况下，自行决定随时增加、修改或重述本文档，增加、修改或重述后的文档对您具有约束力。

使用和披露限制

许可协议

除非移远通信特别授权，否则我司所提供硬软件、材料和文档的接收方须对接收的内容保密，不得将其用于除本项目的实施与开展以外的任何其他目的。

版权声明

移远通信产品和本协议项下的第三方产品可能包含受移远通信或第三方材料、硬软件和文档版权保护的相关资料。除非事先得到书面同意，否则您不得获取、使用、向第三方披露我司所提供的文档和信息，或对此类受版权保护的资料进行复制、转载、抄袭、出版、展示、翻译、分发、合并、修改，或创造其衍生作品。移远通信或第三方对受版权保护的资料拥有专有权，不授予或转让任何专利、版权、商标或服务商标权的许可。为避免歧义，任何形式的购买都不可被视为授予除正常的非独家、免版税的产品使用许可之外的任何许可。对于任何违反保密义务、未经授权使用或以其他非法形式恶意使用所述文档和信息的违法、侵权行为，移远通信有权追究法律责任。

商标

除另行规定，本文档中的任何内容均不授予在广告、宣传或其他方面使用移远通信或第三方的任何商标、商号及名称，或其缩略语，或其仿冒品的权利。

第三方权利

您理解本文档可能涉及一个或多个属于第三方的硬软件和文档（“第三方材料”）。您对此类第三方材料的使用应受本文档的所有限制和义务约束。

移远通信针对第三方材料不做任何明示或暗示的保证或陈述，包括但不限于任何暗示或法定的适销性或特定用途的适用性、平静受益权、系统集成、信息准确性以及与许可技术或被许可人使用许可技术相关的不侵犯任何第三方知识产权的保证。本协议中的任何内容都不构成移远通信对任何移远通信产品或任何其他硬软件、设备、工具、信息或产品的开发、增强、修改、分销、营销、销售、提供销售或以其他方式维持生产的陈述或保证。此外，移远通信免除因交易过程、使用或贸易而产生的任何和所有保证。

隐私声明

为实现移远通信产品功能，特定设备数据将会上传至移远通信或第三方服务器（包括运营商、芯片供应商或您指定的服务器）。移远通信严格遵守相关法律法规，仅为实现产品功能之目的或在适用法律允许的情况下保留、使用、披露或以其他方式处理相关数据。当您与第三方进行数据交互前，请自行了解其隐私保护和数据安全政策。

免责声明

1) 移远通信不承担任何因未能遵守有关操作或设计规范而造成损害的责任。
2) 移远通信不承担因本文档中的任何因不准确、遗漏、或使用本文档中的信息而产生的任何责任。
3) 移远通信尽力确保开发中功能的完整性、准确性、及时性，但不排除上述功能错误或遗漏的可能。除非另有协议规定，否则移远通信对开发中功能的使用不做任何明示、暗示或法定的保证。在适用法律允许的最大范围内，移远通信不对任何因使用开发中功能而遭受的损害承担责任，无论此类损害是否可以预见。
4) 移远通信对第三方网站及第三方资源的信息、内容、广告、商业报价、产品、服务和材料的可访问性、安全性、准确性、可用性、合法性和完整性不承担任何法律责任。

文档历史

修订记录

版本	日期	作者	变更表述
-	2023-09-07	Sharp Sun	文档创建
1.0	2024-07-23	Sharp Sun	受控版本
1.1	2026-01-09	Sharp Sun	添加 PDF 复制代码可能导致非预期换行的说明备注（第 1 章）。

目录

表格索引

图片索引

1 引言

本文档主要介绍如何在移远通信 QSM368ZP-WF 和 SG368Z 系列模块的 Linux 和 Ubuntu 系统上使能 Secure Boot。

备注

由于 PDF 格式特性，文档中部分代码块可能因页面宽度限制自动换行。直接从 PDF 复制代码到编辑器时，可能引入非预期的换行符（\n），导致代码无法正常运行。

建议操作：

若需从 PDF 复制，粘贴后请注意完成如下动作：

- 手动删除复制粘贴后代码中的多余换行符。
- 或通过代码编辑器的替换功能批量删除多余换行符（注意保留语义换行）。

2 Secure Boot 概述

2.1. 简介

Secure Boot 是建立在受信平台上的一种安全启动序列，为用户应用提供一个安全可信的执行环境。为防止任何未经合法签名或恶意修改的软件在设备上运行，Secure Boot 在设备下载和启动过程的每个阶段均增加了签名验证环节，使用加密认证算法校验并加载启动的镜像。

Secure Boot 目的：

- 禁止烧录未经授权的固件；
- 禁止运行未经授权的固件；
- 禁止非法追踪和调试代码；
- 允许安全升级。

2.2. 固件启动序列

设备固件启动序列如下：

1. Maskrom 校验 loader（包含 SPL、ddr 和 usbplug）
2. SPL 校验 *uboot.img*（包含 trust 和 U-Boot 等）
3. U-Boot 校验 *boot.img*（包含 kernel、fdt 和 ramdisk 等）

目前芯片默认只支持 *sha256 + rsa2048 + pkcs-v2.1 (pss) padding* 的安全校验模式。

设备的启动过程分为多个阶段，每个阶段都有一个专用的镜像文件，完成特定的功能。使能 Secure Boot 之后，每一阶段的镜像文件在执行前，都需要前一阶段的镜像文件进行校验，如果校验失败，会造成整个启动过程失败，进而导致设备启动失败。

备注

Secure Boot 必须在工厂阶段开启。禁止通过升级到支持 Secure Boot 的版本进行开启，否则可能会导致部分功能异常且无法恢复。使能 Secure Boot 之前不应该对设备进行配置类的操作，例如，SIMLOCK 配置。

3 Secure Boot 使能步骤

3.1. 环境准备

3.1.1. 安装工具

1. 安装最新版 OpenSSL 工具，用于生成密钥对。
2. 安装 Ubuntu 18.04 或使用 docker 工具加载 Ubuntu 18.04 环境，用于制作 secboot 固件。
3. 安装 RKDevTool 工具，用于烧录 OTP（One-Time-Programmable）文件。
4. 联系移远通信技术支持获取设备对应的 *sec_boot_tools_SGxx.zip* 工具包。该工具包含用于使能 Secure Boot 的相关配置文件。此工具不可单独使用。解压工具包后，需将目录下所有文件拷贝至拆包工具中并覆盖同名文件。

备注

1. 可联系移远通信技术支持（support@quectel.com）协助获取 RKDevTool 工具和安装 OpenSSL、Ubuntu 18.04 及 docker 工具。
2. OTP 指一次性可编译记忆体，通过向其中写入相关数据使能 Secure Boot。

3.2. 生成密钥

在源码根目录下打开 Linux 命令窗口，执行如下命令生成密钥：

```
./build_rk_linux_ubuntu_secboot.sh newkey
```

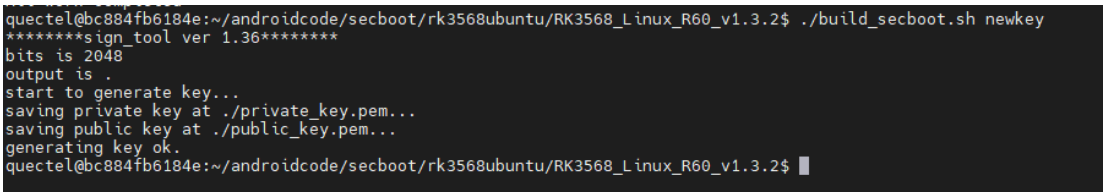


图 1：生成密钥

命令执行成功后，生成的密钥存储在 Linux 源码 *u-boot/keys* 目录下。

备注

1. 如果出现如下类似报错，说明用户目录下没有*.rnd* 文件：

   ```
   Can't load /home4/cjh//.rnd into RNG
   140522933268928:error:2406F079:random number generator:RAND_load_file:Cannot open
   file:../crypto/rand/randfile.c:88:Filename=/home4/cjh//.rnd
   ```

   需先执行 **touch ~/.rnd** 命令手动创建*.rnd* 文件。

2. 上述命令生成的“*keys*”、“*dev.key*”、“*dev.crt*”和“*dev.pubkey*”名称均不可变。因为这些名称已经在 *its* 文件中静态定义，若改变，则会导致打包失败。
3. 设备烧录 OTP 后，此处生成的密钥将不能更改。
4. 密钥只需生成一次，请妥善保管以防丢失或泄漏。
   - 若客户需设置自定义密钥，则可以直接替换在上位机的 *u-boot/keys* 目录下生成的密钥。
   - 后续再次制作 secboot 固件包时，只需将密钥拷贝至上位机的 *u-boot/keys* 目录下。
5. 一旦设备烧录了包含该密钥的 *xxx_loader.bin* 文件使能 Secure Boot 后，则只能使用经过 *u-boot/keys* 路径下的密钥签名的镜像文件。

3.3. 签名镜像

3.3.1. 生成未签名 loader 文件及签名镜像

在 Linux 源码根目录下打开终端窗口，执行如下命令编译，在源码根目录下生成 *secboot_no_loader* 文件夹并生成未签名的 loader 文件及签名镜像：

```
./build_rk_linux_ubuntu_secboot.sh
```

编译结果：

```
.......
//编译完成后，生成未签名的 loader 文件和已签名的 uboot.img 和 boot.img 镜像。
start to sign rk356x_spl_loader_v1.19.112.bin
......
sign loader ok.
......
Image(signed, version=0): uboot.img (FIT with uboot, trust...) is ready
Image(signed, version=0): recovery.img (FIT with kernel, fdt, resource...) is
ready
Image(signed, version=0): boot.img (FIT with kernel, fdt, resource...) is ready
Image(signed): rk356x_spl_loader_v1.15.112.bin (with spl, ddr, usbplug) is ready
pack uboot.img okay! Input: /home4/cjh/rkbin/RKTRUST/RV1126TOS.ini
Platform RV1126 is build OK, with new .config(make rv1126-secure_defconfig)
```

3.3.2. 生成已签名 loader 文件及签名镜像

执行如下命令编译，在源码根目录下生成 *secboot_loader* 文件夹并生成已签名的 loader 文件及签名镜像：

```
./build_rk_linux_ubuntu_secboot.sh sec
```

编译结果：

```
......
//使能 burn-key-hash
### spl/u-boot-spl.dtb: burn-key-hash=1
//编译完成后，生成已签名的 loader 文件和已签名的 uboot.img 和 boot.img 镜像，且包含防回滚版本号。
start to sign rk356x_spl_loader_v1.19.112.bin
......
sign loader ok.
......
Image(signed, version=0, rollback-index=10): uboot.img (FIT with uboot, trust)
is ready
Image(signed, version=0, rollback-index=12): recovery.img (FIT with kernel, fdt,
resource...) is ready
Image(signed, version=0, rollback-index=12): boot.img (FIT with kernel, fdt,
resource...) is ready
Image(signed): rv1126_spl_loader_v1.00.100.bin (with spl, ddr, usbplug) is ready
```

3.4. 烧录 secboot 固件

配置***第 3.3 章***相关步骤后，烧录固件至设备使能 Secure Boot，其中签名的 loader 文件即为 eFuse 文件（用于写入数据至 OTP）。注意，需先烧录未签名的 loader 文件及签名镜像，以确定烧录固件后设备可以正常开机。然后，再烧录签名的 loader 文件及签名镜像使能 Secure Boot。

步骤 1： 打开 RKDevTool 工具，将电脑的 USB 接口与设备的 USB Type-C 口连接，再接通设备电源；按一下“**pwk**”键后松开，进入 loader 模式，若此前未烧录过任何固件版本，设备会直接进入 maskrom 模式进行下载，如下图所示：

图 2：设备进入 loader 模式

步骤 2： 打开 *RKDevTool_Release/RKDevTool.exe* 工具，进行配置。

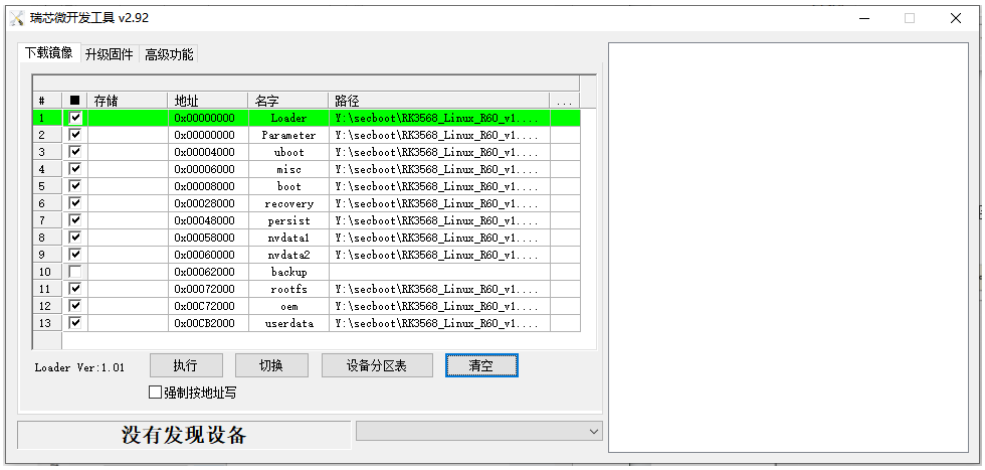


图 3：RKDevTool 工具界面

1) 右键点击设备分区表上方空白区域，选择“**导入配置**”，将编译完成后的 *rockdev/*目录下的 *parameter.txt* 文件中的分区配置加载进来。

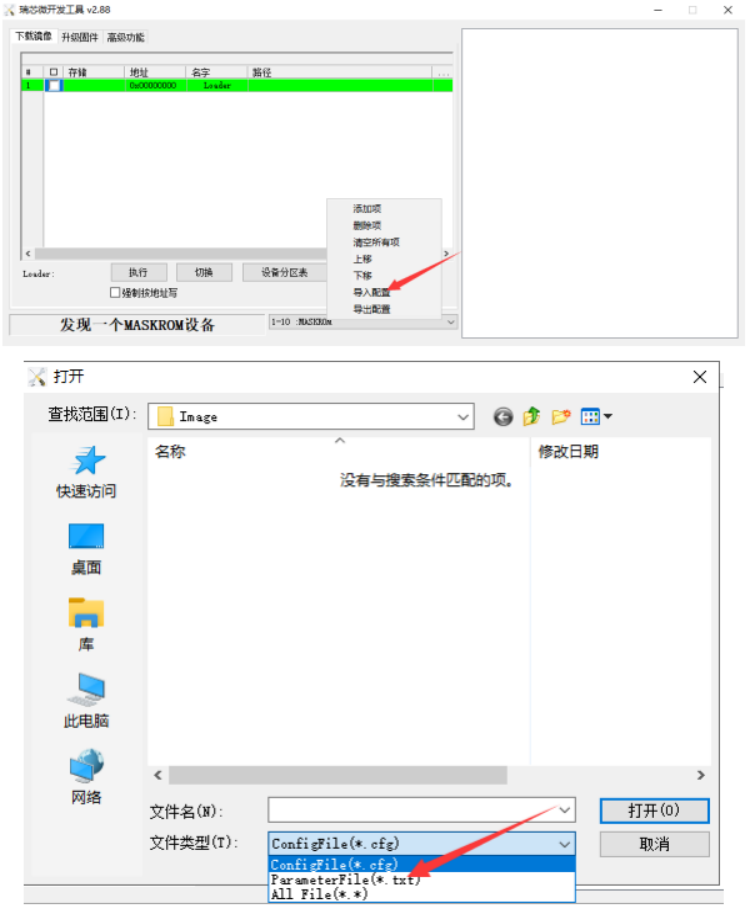
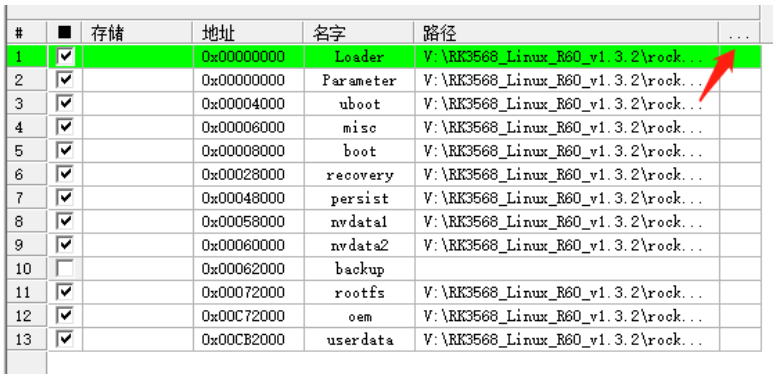


图 4：加载分区配置

2) 加载配置后，根据 *parameter.txt* 文件中的分区名，在下图“**...**”位置选择对应分区镜像。全部选择完成后，选中要烧录的分区，不要选中未选择对应文件的分区，否则执行后会报错，无法开始烧录。



#	■	存储	地址	名字	路径	...
1	☑		0x00000000	Loader	V:\RK3568_Linux_R60_v1.3.2\rock...	
2	☑		0x00000000	Parameter	V:\RK3568_Linux_R60_v1.3.2\rock...	
3	☑		0x00004000	uboot	V:\RK3568_Linux_R60_v1.3.2\rock...	
4	☑		0x00006000	misc	V:\RK3568_Linux_R60_v1.3.2\rock...	
5	☑		0x00008000	boot	V:\RK3568_Linux_R60_v1.3.2\rock...	
6	☑		0x00028000	recovery	V:\RK3568_Linux_R60_v1.3.2\rock...	
7	☑		0x00048000	persist	V:\RK3568_Linux_R60_v1.3.2\rock...	
8	☑		0x00058000	nvdata1	V:\RK3568_Linux_R60_v1.3.2\rock...	
9	☑		0x00060000	nvdata2	V:\RK3568_Linux_R60_v1.3.2\rock...	
10	☐		0x00062000	backup		
11	☑		0x00072000	rootfs	V:\RK3568_Linux_R60_v1.3.2\rock...	
12	☑		0x00C72000	oem	V:\RK3568_Linux_R60_v1.3.2\rock...	
13	☑		0x00CB2000	userdata	V:\RK3568_Linux_R60_v1.3.2\rock...	

图 5：加载镜像

3) 先选择根目录下 *secboot_no_loader* 中的相关文件（即未签名的 loader 文件及签名镜像）进行烧录，确定烧录固件后可以正常开机。然后，再烧录根目录下 *secboot_loader* 中的相关文件（即签名的 loader 文件及签名镜像）使能 Secure Boot。

步骤 3： 点击“**执行**”按钮，开始烧录。

图 6：开始烧录

在右侧空白处会打印 log，显示烧录过程。

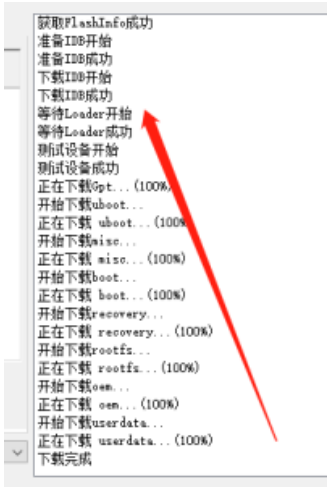


图 7：烧录过程 log

烧录完成后，会自动重启系统。

备注

固件烧录详情，参考***文档 [1]***。

3.5. 检查 Secure Boot 状态

在串口 log 中查看通过 RKDevTool 工具烧录的 OTP 是否成功，判断 Secure Boot 是否使能成功，具体方法如下：

1. 使用 RKDevTool 工具烧录 secboot 固件后，启动设备，保存开机串口 log；
2. 在保存的开机串口 log 中搜索“*Verified-boot:*”；
3. 检查“*Verified-boot:*”后的参数值：
   - “*Verified-boot: 1*”表示 OTP 烧录成功，Secure Boot 使能成功；
   - “*Verified-boot: 0*”表示未烧录过 eFuse 文件，即 Secure Boot 未使能。

4 附录 参考文档及术语缩写

表 1：参考文档

文档名称
[1] Quectel_QSM368ZP-WF&SG368Z 系列_Linux&Ubuntu_编译&烧录指导

表 2：术语缩写

术语	英文全称	中文全称
DDR	Double Data Rate	双倍数据速率
DM	Device Mapper	设备映射
IRAM	Internal Random Access Memory	内部随机访问存储器
OEM	Original Equipment Manufacture	原始设备制造商
OTP	One-Time-Programmable Memory	一次性可编程记忆体
RAM	Random Access Memory	随机存储器
ROM	Read Only Memory	只读存储器
RoT	Root of Trust	可信根
SML	Secure Monitor Layer	安全监控层
SPL	Secondary Program Loader	第二阶段程序加载器
TOS	Trusted Operating System	安全操作系统
USB	Universal Serial Bus	通用串行总线